

patch will update the document. At the next step 1010, the MODIFY message including the patch is transmitted to the collaborator who requested the patch (i.e., the originator of the REQ\_MOD message). Following step 1010, execution returns to step 101 of the method 100.

[0160] A method 1100 of transmitting a PUBLISH message, as executed at step 110 of the method 100, will now be described with reference FIG. 11. The method 1100 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0161] The method 1100 begins at step 1105, where the processor 205 constructs the PUBLISH message from a predetermined opcode. The PUBLISH message includes the username of the collaborator who published the document, a version string corresponding to the document, and the document. At the next step 1110, the PUBLISH message including the document is transmitted to the active collaborators according to the list of active collaborators configured within memory 206, in accordance with a method 1600 shown in FIG. 16. At the next step 1115, the document is emailed to all active collaborators who were offline during the message broadcast of step 1110 (i.e., those collaborators associated with the document who were listed in the list of active collaborators configured within memory 206, but were offline during the broadcast).

[0162] When a collaborator receives a published document by email as at step 1115, the receiving collaborator manually copies the document into a directory of a local hard disk drive (e.g., the hard disk drive 210), where an application incorporating the methods described herein is able to locate the document.

[0163] A method 1200 of transmitting a MODIFY message, as executed at step 112 of the method 100, will now be described with reference FIG. 12. The method 1200 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0164] The method 1200 begins at step 1205, where the processor 205 constructs the MODIFY message from predetermined opcode. The MODIFY message includes the username of the collaborator who created the patch, a version string corresponding to the document to which the patch applies (i.e., the document which was updated), a further version string corresponding to the version to which the patch will update the document, the patch and the document UUID. At the next step 1210, the MODIFY message including the patch is transmitted to the active collaborators according to the list of active collaborators configured within memory 206, in accordance with a method 1600 shown in FIG. 16.

[0165] The method 1300 of updating a document, as executed at step 324, will now be described with reference to FIG. 13. The method 1300 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0166] The method 1300 begins at step 1305, where if the processor 205 determines that the originator of the MODIFY message, detected at step 301, participated in the editing of the current version of the document stored in the memory 206, then the method 1300 proceeds to step 1310. Otherwise

the method 1300 proceeds to step 1315. At step 1310, the modifications represented by the patch accompanying the MODIFY message are applied to the current version of the document stored in memory 206. Following step 1310, execution proceeds to step 326.

[0167] At step 1315, the processor 205 determines a patch representing the difference between the current version of the document stored in memory 206 and the version of the document (i.e., the version of the document represented by the version string) that has been identified in the MODIFY message. Then at the next step 1320, the processor 205 merges the documents by merging the patch that was received with the MODIFY message with the patch determined at step 1315. A method 1400 of merging two patches representing modifications to the document will be described in detail below with reference to FIG. 14. The method 1300 concludes at the next step 1325, where the merged set of document modifications is applied to the current document stored in memory 206. Following step 1325, execution returns to step 326.

[0168] The method 1400 of merging two patches representing modifications to the document will now be described with reference to FIG. 14. The method 1400 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0169] The method 1400 begins at step 1405, where if the processor 205 determines that there is a component (i.e., a current component) of the merged patch that needs to be determined, then the method 1400 proceeds to step 1410. Otherwise, the method 1400 concludes and execution returns to step 1325 of the method 1300.

[0170] At step 1410, the processor 205 determines which patch modified the current component identified at step 1405. If neither the patch determined at step 1315 or the patch accompanying the MODIFY message modified the current component, then the method 1400 returns to step 1405.

[0171] If the patch determined at step 1315 was responsible for the modification to the current component then the method 1400 proceeds to step 1415. At step 1415, the component value corresponding to the current component of the patch determined at step 1315 is used to modify the component value for the merged document.

[0172] If the patch accompanying the MODIFY message detected at step 301 was responsible for the modification to the current component identified at step 1415, then the method 1400 proceeds to step 1420. At step 1415, the component value corresponding to the current component of the patch accompanying the MODIFY message is used to modify the component value for the current component of the merged document.

[0173] If both the patch determined at step 1315 and the patch accompanying the MODIFY message modified the current component identified at step 1405, then the method 1400 proceeds to step 1425. At step 1425, the processor 205 determines how to modify the current component.

[0174] If sequential modifications to the current component need to be added, then execution continues at step 1430. At step 1430, the value of the current component is set to the weighted arithmetic mean of the component value corre-